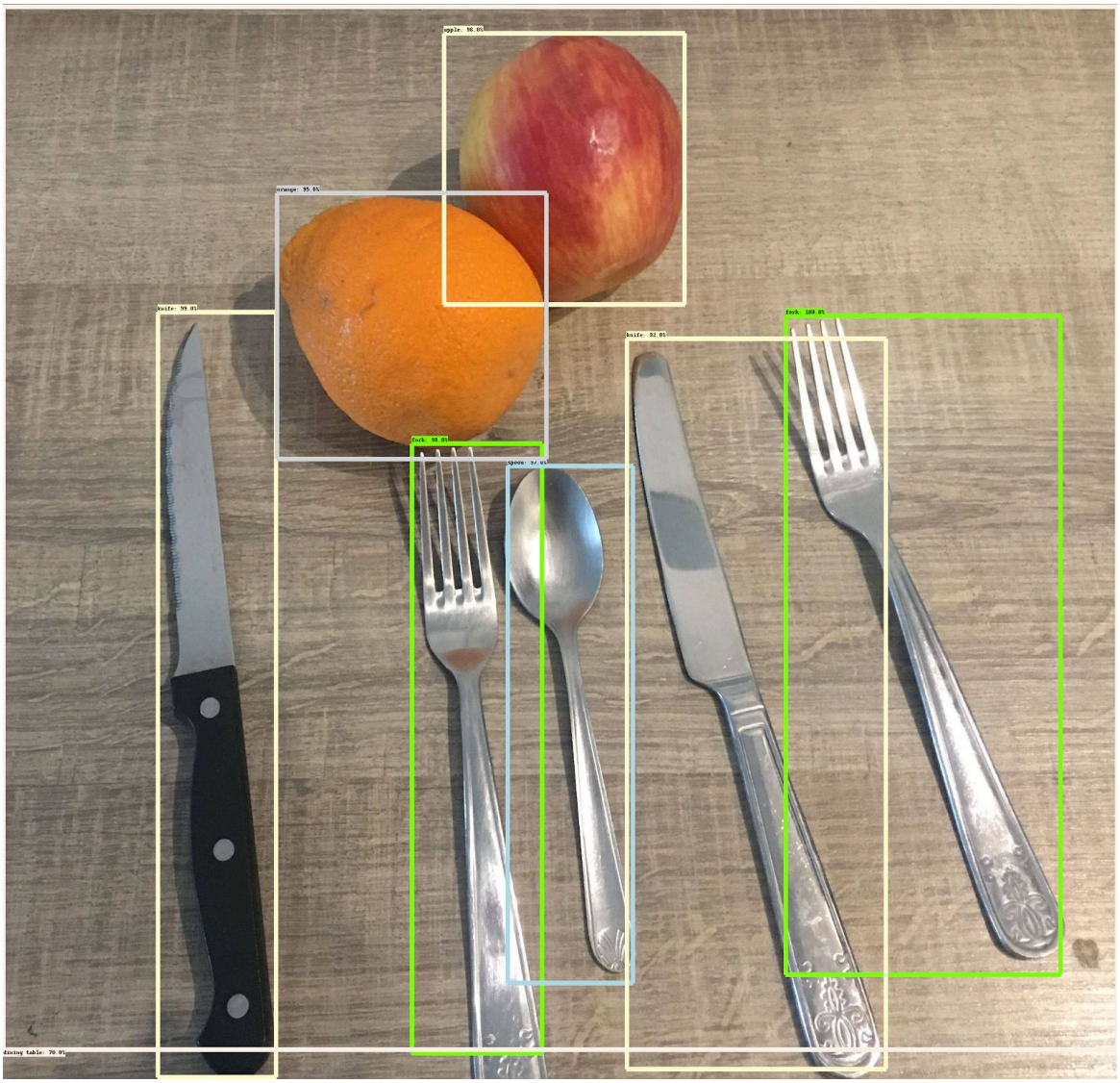
Rendu TP détection d'objets:

Rédigé par :
-Nada El Abdellaoui
-Dina Abakkali

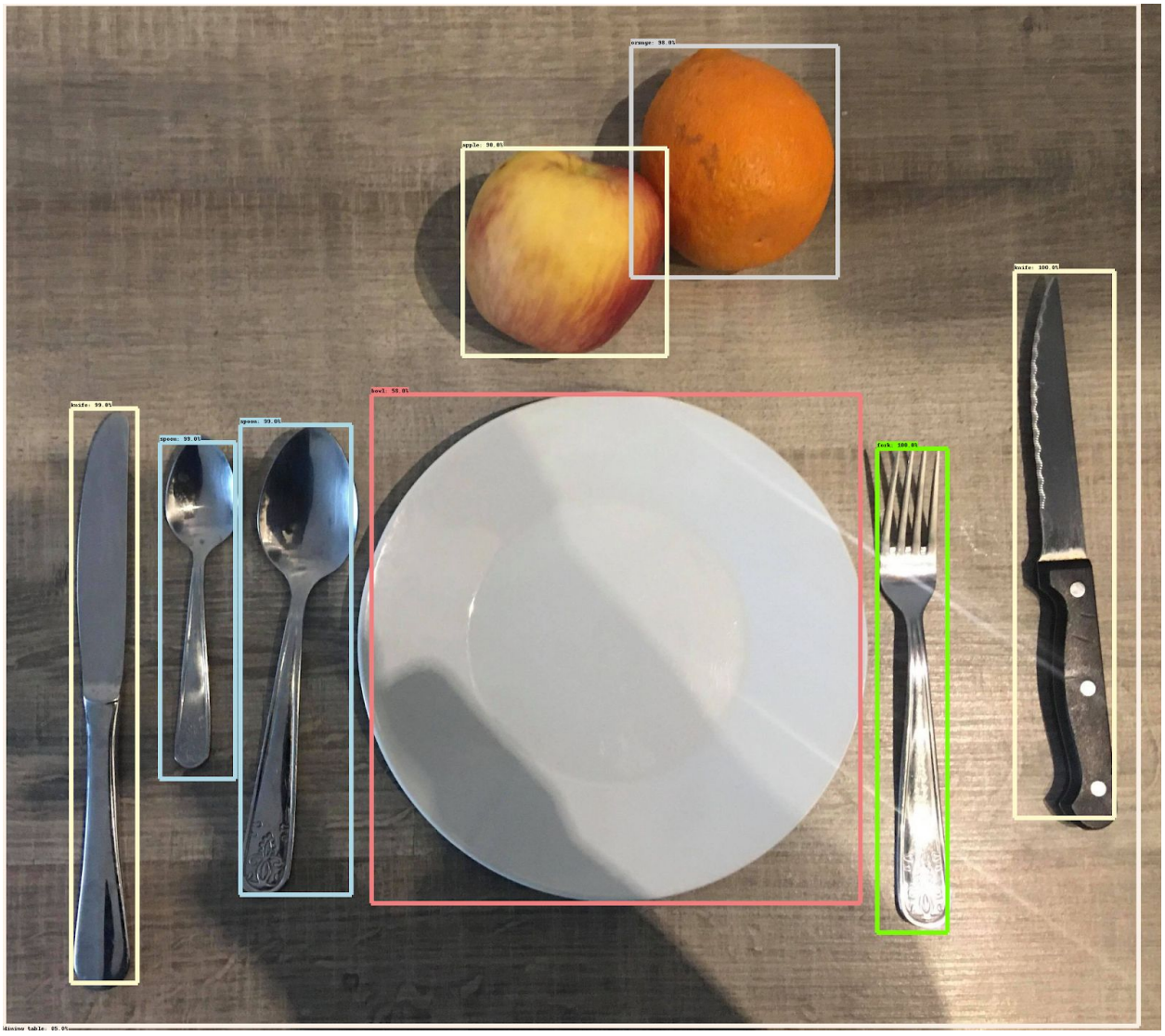
1. Détection objet avec la méthode Fast(er)-RCNN :

Pour cette partie , on a fait des recherches et on a trouvé des codes déjà prêts, mais qui utilisaient d'autres modèles que Faster RCNN.
Du coup, on a été amené à faire les modifications nécessaires pour que la détection soit faite par Faster RCNN.

Dans la première photo , on remarque qu'on a pu détecter la fourchette , le couteau , la cuillère et les fruits (Pomme et Orange) avec des hautes précisions.



Dans la 2ème photo , nous avons changé la position des éléments et nous avons ajouté un plat.



2. Ajout d’une classe qui ne figure pas sur COCO:

Pour cette partie , nous avons choisi d’ajouter deux classes qui ne figure pas sur la dataset (COCO) :
Classes KIWI et EGG.

Les étapes pour un fine-tuning :

A- Importations:

On commence d’abord par importer toutes les bibliothèques nécessaires.

B-Utilitaires

On ajoute les fonctions de chargement d’images et on les charge.

C-Annoter des images avec des bounding boxes

Dans cette partie, on annote les “Kiwi” et “Egg” en dessinant des bounding boxes autours des kiwi et des oeufs sur chaque image.(on a travaillé sur chaque classe séparément) . On clique sur “next image” pour passer à l’image suivante et on clique sur “submit” lorsqu’on a terminé .

D-Préparer les données pour l'apprentissage

On ajoute les annotations de classe (pour simplifier on travaille sur une seule classe à la fois) et on convertit les bounding boxes et les étiquettes en tenseurs et on encode les étiquettes.

E-Créez un modèle et restaurez les poids pour toutes les couches sauf la dernière

F-Boucle d'entraînement personnalisée en mode Eager

G-Charger les images de test et exécuter l'inférence avec un nouveau modèle.

Pour les 3 dernières étapes on n’a pas pu travailler dessus , vu qu’on avait pas assez de temps .

Pour compléter le travaille il fallait :

- Télécharger le modèle et le charger en mémoire (on l’a déjà fait)
- Créer un model de détection
- Configurer la restauration des object-based checkpoints
- Exécuter le modèle à travers une image dummy afin de créer les variables image et shapes
- Réentraîner le modèle à partir d’une boucle d'entraînement personnalisé
- Et enfin, choisir les variables de la couche supérieure pour le fine-tuning.

```
dummy_scores = np.array([1.0], dtype=np.float32)  # give boxes a score of 100%

plt.figure(figsize=(30, 15))
for idx in range(0,7):
  plt.subplot(4, 3, idx+1)
  plot_detections(
      train_images_np[idx],
      gt_boxes[idx],
      np.ones(shape=[gt_boxes[idx].shape[0]], dtype=np.int32),dummy_scores, category_index)
plt.show()
```